

CLAIMS

1. A method for managing the availability of compiled data in a logically partitioned computer system having a plurality of processors, comprising the steps of:

defining a plurality of logical partitions of said computer system, each logical partition having a respective processor resource assignment, wherein each task executing in said computer system is assigned to a respective one of said logical partitions and wherein the definition of a plurality of logical partitions may be dynamically altered;

automatically determining first and second discrete subsets of said compiled data within a first logical partition, wherein said first discrete subset of said compiled data can be rebuilt using a first set of resources assigned to said first logical partition within at least one predetermined recovery time limit;

automatically storing said second subset of said compiled data in a form which does not require rebuilding of said second subset;

automatically detecting a configuration change to said first logical partition which alters the processor resource assignment of said first logical partition;

automatically determining third and fourth discrete subsets of said compiled data within said first logical partition, wherein said third discrete subset of said compiled data can be rebuilt using a second set of resources assigned to said first logical partition within said at least one predetermined recovery time limit, said second set of resources being the set of resources after assigned to said first logical partition after said configuration change, said second set of resources being different from said first set of resources; and

thereafter automatically storing said fourth subset of compiled data in a form which does not require rebuilding of said fourth subset.

2. The method for managing the availability of compiled data of claim 1, wherein said compiled data comprises a plurality of database indexes.

3. The method for managing the availability of compiled data of claim 1, wherein said predetermined recovery time limit is a user specified recovery time limit.

4. The method for managing the availability of compiled data of claim 1, wherein said step of automatically storing said second subset of compiled data comprises logging changes to said second subset of compiled data in at least one non-volatile storage device, and wherein said step of automatically storing said fourth subset of compiled data comprises logging changes to said fourth subset of compiled data in at least one non-volatile storage device.

5. The method for managing the availability of compiled data of claim 1, wherein:
 if said step of automatically detecting a configuration change to said first logical partition detects a configuration change which increases the processor resource assignment of said first logical partition, then said step of automatically determining third and fourth discrete subsets determines a fourth subset which is a subset of said second subset; and
 if said step of automatically detecting a configuration change to said first logical partition detects a configuration change which reduces the processor resource assignment of said first logical partition, then said step of automatically determining third and fourth discrete subsets determines a third subset which is a subset of said first subset.

10074889-100904

1 6. The method for managing the availability of compiled data of claim 1,
2 wherein said step of automatically determining first and second discrete subsets of
3 said compiled data comprises defining a plurality of buckets for said compiled data, each
4 bucket corresponding to a respective rebuild time; determining a respective cumulative
5 rebuild time for each bucket, said cumulative rebuild time representing the rebuild time using
6 said first set of resources of all compiled data in the respective bucket and in all buckets
7 corresponding to smaller rebuild times; and selecting the compiled data in all buckets having
8 cumulative rebuild times less than said predetermined recovery time limit for inclusion in
9 said first discrete subset; and
10 wherein said step of automatically determining third and fourth discrete subsets of
11 said compiled data comprises determining a respective cumulative rebuild time for each
12 bucket, said cumulative rebuild time representing the rebuild time using said second set of
13 resources of all compiled data in the respective bucket and in all buckets corresponding to
14 smaller rebuild times; and selecting the compiled data in all buckets having cumulative
15 rebuild times less than said predetermined recovery time limit for inclusion in said third
16 discrete subset.

1 7. The method for managing the availability of compiled data of claim 6,
2 wherein each said bucket comprises at least one sub-bucket, each sub-bucket
3 corresponding to a processor resource;
4 wherein if said second set of resources is greater than first set, then said step of
5 automatically determining third and fourth discrete subsets comprises allocating additional
6 sub-buckets corresponding to the difference between said first set of resources and said
7 second set of resources, and transferring compiled data units from previously existing sub-
8 buckets to said additional sub-buckets; and
9 wherein if said second set of resources is less than said first set, then said step of
10 automatically determining third and fourth discrete subsets comprises removing sub-buckets
11 corresponding the difference between said first set of resources and said second set of
12 resources, and transferring compiled data units from the removed sub-buckets to remaining
13 sub-buckets.

1 8. A program product for managing the availability of compiled data in a computer
2 system having a plurality of central processors and a dynamic logical partitioning
3 mechanism, said dynamic logical partitioning mechanism supporting a plurality of defined
4 logical partitions of said computer system, each logical partition having a respective
5 processor resource assignment, wherein each task executing in said computer system is
6 assigned to a respective one of said logical partitions and wherein the definition of a plurality
7 of logical partitions may be dynamically altered, said program product comprising a plurality
8 of processor executable instructions recorded on signal-bearing media, wherein said
9 instructions, when executed by at least one central processor of said computer system, cause
10 the system to perform the steps of:

11 automatically determining first and second discrete subsets of said compiled data
12 within a first logical partition, wherein said first discrete subset of said compiled data can be
13 rebuilt using a first set of resources assigned to said first logical partition within at least one
14 predetermined recovery time limit;

15 automatically storing said second subset of said compiled data in a form which does
16 not require rebuilding of said second subset;

17 automatically detecting a configuration change to said first logical partition which
18 alters the processor resource assignment of said first logical partition;

19 automatically determining third and fourth discrete subsets of said compiled data
20 within said first logical partition, wherein said third discrete subset of said compiled data can
21 be rebuilt using a second set of resources assigned to said first logical partition within said
22 at least one predetermined recovery time limit, said second set of resources being the set of
23 resources after assigned to said first logical partition after said configuration change, said
24 second set of resources being different from said first set of resources; and

25 thereafter automatically storing said fourth subset of compiled data in a form which
26 does not require rebuilding of said fourth subset.

1 9. The program product for managing the availability of compiled data of claim 8,
2 wherein said compiled data comprises a plurality of database indexes.

13. The program product for managing the availability of compiled data of claim 8, wherein said step of automatically determining first and second discrete subsets of said compiled data comprises defining a plurality of buckets for said compiled data, each bucket corresponding to a respective rebuild time; determining a respective cumulative rebuild time for each bucket, said cumulative rebuild time representing the rebuild time using said first set of resources of all compiled data in the respective bucket and in all buckets corresponding to smaller rebuild times; and selecting the compiled data in all buckets having cumulative rebuild times less than said predetermined recovery time limit for inclusion in said first discrete subset; and

wherein said step of automatically determining third and fourth discrete subsets of said compiled data comprises determining a respective cumulative rebuild time for each bucket, said cumulative rebuild time representing the rebuild time using said second set of resources of all compiled data in the respective bucket and in all buckets corresponding to smaller rebuild times; and selecting the compiled data in all buckets having cumulative rebuild times less than said predetermined recovery time limit for inclusion in said third discrete subset.

1 14. The program product for managing the availability of compiled data of claim 13,
2 wherein each said bucket comprises at least one sub-bucket, each sub-bucket
3 corresponding to a processor resource;
4 wherein if said second set of resources is greater than first set, then said step of
5 automatically determining third and fourth discrete subsets comprises allocating additional
6 sub-buckets corresponding to the difference between said first set of resources and said
7 second set of resources, and transferring compiled data units from previously existing sub-
8 buckets to said additional sub-buckets; and
9 wherein if said second set of resources is less than said first set, then said step of
10 automatically determining third and fourth discrete subsets comprises removing sub-buckets
11 corresponding the difference between said first set of resources and said second set of
12 resources, and transferring compiled data units from the removed sub-buckets to remaining
13 sub-buckets.

14 15. A computer system, comprising:
15 a plurality of central processing units;
16 a memory;
17 a logical partitioning mechanism supporting a plurality of defined logical partitions of
18 said computer system, each logical partition having a respective processor resource
19 assignment, wherein each task executing in said computer system is assigned to a respective
20 one of said logical partitions and wherein the definition of said logical partitions may be
21 dynamically altered;
22 a protection utility for compiled data in a first logical partition of said computer
23 system, wherein said protection utility:
24 (a) automatically computes rebuild times of discrete portions of said compiled
25 data according to a first configuration of said first logical partition to determine a first
26 subset of said compiled data to be stored in a form which does not require rebuilding,
27 said first subset being determined to meet a predetermined recovery time limit;
28 (b) automatically detects a configuration change to said first logical partition
29 which affects the time required to rebuild said compiled data, said configuration
30 change configuring said first logical partition according to a second configuration;
31 (c) responsive to detecting said configuration change, computes rebuild
32 times of said discrete portions of said compiled data according to said second
33 configuration of said first logical partition; and
34 (d) automatically determines a second subset of said compiled data to be
35 stored in a form which does not require rebuilding using said rebuild times of said
36 discrete portions of said compiled data according to said second configuration of said
37 first logical partition, said second subset being determined to meet said predetermined
38 recovery time; and
39 (e) automatically migrates compiled data stored in said form which does not
40 require rebuilding from said first subset to said second subset.

1 16. The computer system of claim 15, wherein said compiled data comprises a plurality
2 of database indexes.

1 17. The computer system of claim 15, wherein said form which does not require
2 rebuilding comprises one or more log records stored in a non-volatile storage device.

1007449-10004
1007449-10004